



JOINT POLAR SATELLITE SYSTEM (JPSS) COMMON GROUND SYSTEM (CGS)

IDPS PRO SOFTWARE CONFIGURATION GUIDE USER MANUAL

CDRL No. A032

**RAYTHEON COMPANY
INTELLIGENCE AND INFORMATION SYSTEMS (IIS)
JPSS CGS PROGRAM
AURORA, COLORADO**

**Copyright © 2011
Raytheon Company
Unpublished Work
ALL RIGHTS RESERVED**

This data was developed pursuant to Contract Number NNG10XA03C with the US Government. The US government's rights in and to this copyrighted data are as specified in DFAR 252.227-7013 which was made part of the above contract.

HARDCOPY UNCONTROLLED

IAW DFARS 252.227-7036, Raytheon hereby declares that, to the best of its knowledge and belief, the technical data delivered under Contract No. NNG10XA03C is complete, accurate, and complies with all requirements of the Contract.

TITLE: JOINT POLAR SATELLITE SYSTEM (JPSS) COMMON GROUND SYSTEM (CGS)
IDPS PRO SOFTWARE CONFIGURATION GUIDE USER MANUAL

APPROVAL SIGNATURES:

(Signatures indicate compliance with these requirements)

William
Sullivan

Digitally signed by William Sullivan
DN: cn=William Sullivan, o=JPSS,
ou=DCMS,
email=wjsullivan@raytheon.com, c=US
Date: 2011.12.09 12:53:43 -07'00'

William J Sullivan
Program Manager

Date

Frank Zorn

Digitally signed by Frank Zorn
DN: cn=Frank Zorn, o=JPSS
Sustainment, ou=JPSS Sustainment,
email=fgzorn@raytheon.com, c=US
Date: 2011.12.08 10:26:17 -07'00'

Frank Zorn
Sustainment IPT Lead

Date

Harvey Lee

Digitally signed by Harvey Lee
DN: cn=Harvey Lee, o=JPSS,
ou=JPSS,
email=hylee@raytheon.com, c=US
Date: 2011.12.01 15:16:10 -07'00'

Harvey Y. Lee
IDPS Sustainment Lead

Date

Justin Lee

Digitally signed by Justin Lee
DN: cn=Justin Lee, o=IDPS Sustainment,
ou=JPSS,
email=justin.lee@raytheon.com, c=US
Date: 2011.12.01 12:54:02 -07'00'

Justin E Lee
Software Manager

Date

Sheryl
Fertman

Digitally signed by Sheryl Fertman
DN: cn=Sheryl Fertman, o=IIS QA, ou=IIS
QA, email=sfertman@raytheon.com,
c=US
Date: 2011.12.03 09:01:43 -07'00'

Sheryl Fertman
IIS Quality

Date

HARDCOPY UNCONTROLLED

Revision/Change History

Revision	Document Date	Revision/Change Description	Pages Affected
-	07 Jul 2011	Initial Release per ECR-IDP-3733	All
A	30 Nov 2011	Released per ECR-IDP-004062	All

HARDCOPY UNCONTROLLED

Table of Contents

1	SCOPE	1
1.1	Identification	1
1.2	System Overview.....	1
1.2.1	Common Ground System Overview	1
1.2.1.1	IDPS Overview	1
1.3	Document Overview	2
2	REFERENCE DOCUMENTS.....	3
3	SOFTWARE CONFIGURATION SUMMARY	4
3.1	Guide Lists.....	4
3.2	Configuration Guides.....	4
3.2.1	General Processing Configuration Identifiers	4
3.2.1.1	Algorithm - Group Identifier.....	5
3.2.1.1.1	Version - Name Identifier.....	5
3.2.1.2	ConfigValues - Group Identifier	5
3.2.1.3	Duration - Group Identifier	5
3.2.1.3.1	Time - Name Identifier.....	6
3.2.1.4	Inputs - Group Identifier	6
3.2.1.4.1	ProCmnShortNameValue – Group Identifier	6
3.2.1.4.1.1	OfficialShortName_X – Name Identifier	7
3.2.1.4.1.2	SubstituteShortName_X – Name Identifier	7
3.2.1.4.1.3	OfficialAlgBranch_X – Name Identifier	8
3.2.1.4.1.4	SubstituteAlgBranch_X – Name Identifier	8

HARDCOPY UNCONTROLLED

3.2.1.4.1.5SubstituteShortNameConversionProduct_X – Name Identifier	9
3.2.1.4.1.6Mode – Name Identifier	9
3.2.1.4.1.7Wait – Name Identifier.....	10
3.2.1.4.1.8Crossgranule_Scans – Name Identifier.....	10
3.2.1.4.1.9DataEndianType – Name Identifier	10
3.2.1.5 Outputs - Group Identifier	11
3.2.1.5.1 ProCmnShortNameValue – Group Identifier	11
3.2.1.5.1.1OfficialShortName – Name Identifier.....	11
3.2.1.5.1.2SubstituteShortName – Name Identifier.....	11
3.2.1.5.1.3Mode	12
3.2.1.5.1.4OutputType – Name Identifier	12
3.2.1.5.1.5DataEndianType – Name Identifier	13
3.2.1.5.2 N_LEOA_Flag – Metadata Set Flag	14
3.2.2 Controller Algorithms.....	14
3.2.2.1 SubAlgorithms - Group Identifier	14
3.2.2.1.1 Algorithm_X – Group Identifier	14
3.2.2.1.1.1AlgorithmName – Name Identifier	14
3.2.2.1.1.2Params – Group Identifier	15
3.2.2.1.1.2.1 Param_X – Name Identifier.....	15
3.2.3 GridToGran Algorithm	16
3.2.3.1 EnvironmentRunName – Name Identifier	16
3.2.3.2 Constellation – Name Identifier.....	16
3.2.4 GranToGrid Algorithm	16

HARDCOPY UNCONTROLLED

3.2.4.1	AllowGracefulDegradationFlag – Name Identifier	17
3.2.4.2	AllowSpacecraftManeuversFlag – Name Identifier	17
3.2.4.3	ProGipPseudo – Name Identifier	17
3.2.4.4	Constellation – Name Identifier.....	17
3.2.5	Granulate Ancillary Algorithm	17
3.2.5.1	EnvironmentRunName – Name Identifier	18
3.2.6	Granulate Ancillary Controller	18
3.2.7	WCalc Algorithm.....	18
3.2.7.1	NumberOfDetectors – Name Identifier.....	18
3.2.7.2	AggregationZones – Group Identifier.....	18
3.2.7.2.1	AggZone_X – Group Identifier.....	19
3.2.7.2.1.1	StartPixelColumn – Name Identifier	19
3.2.7.2.1.2	NumPixelsInZone – Name Identifier	19
3.2.7.2.1.3	SmearFactor – Name Identifier	19
APPENDIX A – EXAMPLE OF GENERAL ALGORITHM CONFIGURATION GUIDE...		20
APPENDIX B – EXAMPLE CONTROLLER ALGORITHM CONFIGURATION GUIDE .		23
APPENDIX C – EXAMPLE GRANULATE ANCILLARY-SPECIFIC CONTROLLER ALGORITHM CONFIGURATION GUIDE		26
APPENDIX D – EXAMPLE GRAN ANC ALGORITHM CONFIGURATION GUIDE.....		33
APPENDIX E – EXAMPLE WCALC ALGORITHM CONFIGURATION GUIDE.....		34
APPENDIX G – EXAMPLE USING NOAUTOSSETUP MODE (FROM GTM IMAGERY MBAND CONFIGURATION GUIDE)		37
APPENDIX H – EXAMPLE USING NOAUTOSSETUP MODE FOR OUTPUTS (FROM VIIRS SDR ATMSREMAP CONFIGURATION GUIDE)		38

HARDCOPY UNCONTROLLED

List of Tables

Table 2-1 Referenced Documents	3
--------------------------------------	---

1 SCOPE

The Interface Data Processor Segment (IDPS) PRO Software User's Manual Part I describes the architecture, installation and setup of the Processing Subsystem (PRO). This document has been prepared in support of the PRO Software Item (SI), of the Joint Polar- Satellite System Common Ground System (JPSS-CGS) and the NPOESS Preparatory Project (NPP).

1.1 Identification

The IDPS Processing Software Configuration Guide User Manual describes the configuration of the Processing Subsystem (PRO).

1.2 System Overview

The Joint Polar Satellite System (JPSS) is a System-of-Systems architecture whose mission is to provide military and civilian agencies with environmental, meteorological, and climatological data and products. The Common Ground System (CGS) provides command and control of the National Polar-orbiting Operational Environmental Satellite System (NPOESS) Preparatory Project (NPP) and JPSS spacecraft and instruments. The CGS transports commands, telemetry, and mission data between the space, ground control, and processing facilities and the Customer locations, and enables mission planning, asset management, and generation and delivery of the mission products.

1.2.1 Common Ground System Overview

The CGS Program is responsible for the design, implementation, integration, test, deployment, and transition of the three JPSS Ground Segments: the Command, Control, and Communications Segment (C3S), the Interface Data Processing Segment (IDPS), and the Field Terminal Segment (FTS). C3S manages the overall JPSS CGS mission, including active management and accounting of mission data, from storage on the satellite solid-state recorder to delivery to the IDPS. The IDPS is responsible for receiving raw mission data from C3S and for creating and delivering useable environmental products to Users. FTS provides hardware and software specifications needed to build JPSS CGS-compliant field terminals (FTs), along with updated processing software designed to handle direct broadcast data streams.

1.2.1.1 IDPS Overview

There is an Interface Data Processor (IDP) deployed at each Operational Central, located at National Environmental Satellite, Data, and Information Service (NESDIS), Air Force Weather Agency (AFWA), Fleet Numerical Meteorology and Oceanography Center (FNMOC), and Naval Oceanographic Office (NAVO). The primary function of each IDP is to ingest and process Stored Mission Data (SMD) into data products and then to deliver the data products to internal and external users and systems. The IDP provides the data products to the Centrals based on system level latency requirements. Additionally, the IDP supports long term monitoring, calibration and validation of the products and satellite sensors.

The IDP ingests and processes SMD received from the Command, Control, and Communications Segment (C3S). The artifacts from satellite on-board storage and ground communication routing are removed and the resulting Application Packets (APs) are then stored as Raw Data Records (RDRs) as raw data on a per sensor and spacecraft basis. Each IDP also ingests Auxiliary and Ancillary data needed in the processing of the SMD. The RDRs, Auxiliary data, and Ancillary data are subsequently processed to create Sensor Data Records (SDRs), Intermediate Products (IPs), Application Related Products (ARPs), and Environmental Data Records (EDRs). The created data products, excluding RDRs, are normally geolocated with the geospatial information. Once the data records are created, they are delivered, along with the a subset of the Auxiliary and Ancillary data, as requested to the users, including the Centrals, the NOAA Science Investigator-led Processing System (NSIPS), the C3S Payload Support Tool (PST), the NASA Science Data Segment (SDS), and the NOAA Comprehensive Large Array-data Stewardship System (CLASS).

1.3 Document Overview

This document defines the Configuration Guides of the IDPS Processing (PRO) Subsystem.

2 REFERENCE DOCUMENTS

This section identifies the documents referenced in this document.

Table 2-1 Referenced Documents

Document #	Document Title
UG60917-IDP-011	IDPS INF SW User Manual
474-00001-01	JPSS Common Data Format Control Book – External Volume I - Overview
LI60917-GND-005	JPSS CGS Acronyms & Glossary

HARDCOPY UNCONTROLLED

3 SOFTWARE CONFIGURATION SUMMARY

3.1 Guide Lists

Guide Lists are files that provide a list of all configuration files needed by an executable to be able to run. The configuration files are not limited to just PRO configuration but include segment and additional subsystem level configuration files.

3.2 Configuration Guides

This section contains information on identifiers placed in the PRO configuration guides. PRO uses the configuration guide for determining collection short names to be used when querying DMS for given items, controlling program execution, values to be used when populating metadata, and controlling output. A configuration guide consists of group name identifiers, which denote groups of configuration items and name identifiers containing actual configuration values. All configuration identifiers and configuration values are case sensitive so developers should pay special attention to case when developing configuration guides. Below is an example showing the group name and name identifiers that would be found in a configuration guide.

```
<group name="Group_Name_Value">
  <config>
    <name>Name_Value_1</name>
    <configValue>String_Value_1</configValue>
  </config>
  <config>
    <name>Name_Value_2</name>
    <configValue>String_Value_2</configValue>
  </config>
</group>
```

The group name and name identifiers are the building blocks for all PRO configuration guides. As new configuration guide entries are added into the PRO software, these identifiers should be added to this document. New changes to the configuration guide document should be reviewed at the design review which is making the additions to this document.

The following sections provide explanations for identifiers used in PRO configuration guides. Finally, example configuration guides are shown in the appendices.

There should be one configuration guide per derived algorithm and configuration guides in processing should be named to match the Algorithm name. For example the ProSdrViirs derived software would have a configuration guide named ProSdrViirs_CFG.xml.

3.2.1 General Processing Configuration Identifiers

At the top of the hierarchy in PRO configuration guides is the algorithm name. The algorithm name is what separates one algorithm's configuration guide from another and localizes the information to that algorithm. The algorithm name was previously named the routine name and set via the Common Algorithm functionality.

HARDCOPY UNCONTROLLED

This is the first entry in a configuration guide and as PRO looks up other parameters in the configuration guide it will start each search with:

Algorithm Name.XXX

where XXX is the next group identifier or configuration name value.

The following subsections will describe configuration values which are general to all configuration guides. An example general configuration guide is shown in Appendix A.

3.2.1.1 Algorithm - Group Identifier

The *Algorithm* group identifier contains the following configuration values. The *Algorithm* configuration items are required of all algorithms.

<Algorithm Name>.**Algorithm**

3.2.1.1.1 Version - Name Identifier

The *Version* name identifier is also used to build the algorithm version string for populating metadata for output products. The *Version* value is concatenated to the end of the *ID* string to form the algorithm version string. It is read from the configuration guide during algorithm initialization.

<Algorithm Name>.<Algorithm>.**Version**

The *Version* is populated with a value of "1.O.001.001" for all configuration guides starting in the Maintenance build I1.5.05.00. See Volume I of the CDFCB-X in section 3.4.8 for future version definitions.

3.2.1.2 ConfigValues - Group Identifier

The *ConfigValues* group identifier contains the specific configuration values needed by an algorithm to control an individual GridToGran algorithm.

<Algorithm Name>.**ConfigValues**

3.2.1.3 Duration - Group Identifier

The *Duration* group identifier is used to specify a duration time to be used when querying DMS for inputs. The PRO software looks for this group identifier in the configuration guide prior to retrieving any input. If no *Duration* is specified, then the default behavior is that 'no time duration will occur for queries to DMS'.

<Algorithm Name>.**Duration**

3.2.1.3.1 Time - Name Identifier

The *Time* name identifier is used to specify the duration time, in seconds, to wait before declaring failure on a given input. Input items need to have a configuration entry of *Wait* (explained in Section 3.2.1.4.1.7) for each item in which a wait should be performed. If the *Wait* entry is not specified, the effect will be that the duration time will be ignored and the query will immediately fail.

<Algorithm Name>.Duration.**Time**

The *Time* is currently configured with a numerical string value denoting the number of seconds to wait for an input. For example, 120 would indicate a two minute duration time.

3.2.1.4 Inputs - Group Identifier

The *Inputs* group identifier contains, underneath it, the list of inputs to be used by the algorithm to map string names defined in the common software to values for the metadata identifier, *N_Collection_Short_Name*, for retrieving inputs. This section lists inputs to be used by an algorithm. An algorithm may not use all inputs listed; for instance a controller algorithm could put an input on a map for an algorithm, but any input that is retrieved by an algorithm will be listed in this section.

<Algorithm Name>.**Inputs**

The *Inputs* group identifier can be followed by an optional spacecraft identifier to differentiate inputs specific to a particular spacecraft. For example:

<Algorithm Name>.**Inputs.NPP**

In the following subsections the spacecraft identifier will be denoted as [Spacecraft] as a potentially optional configuration item.

3.2.1.4.1 ProCmnShortNameValue – Group Identifier

The *ProCmnShortNameValue* group identifier value is used to map to an actual collection short name value to be used in the query of DMS. The *ProCmnShortNameValue* string should not be confused with an actual collection short name value as defined in the DPIS, but rather is merely a string which maps to an actual collection short name. Due to the inappropriate naming of the header and source files this has, and can be, a source of confusion.

There are multiple entries of *ProCmnShortNameValues* under the *Inputs* group identifier and *ProCmnShortNameValue* itself is also a group identifier.

<Algorithm Name>.**Inputs.[Spacecraft.]ProCmnShortNameValue**

3.2.1.4.1.1 OfficialShortName_X – Name Identifier

The *OfficialShortName_X* name identifier lists out collection short names, in priority order, to be used for retrieval and graceful degradation; where the *X* represents a number. For instance, the *OfficialShortName_1* entry will specify the primary collection short name value to be used in the DMS query. *OfficialShortName_2* entry will specify the secondary or alternate collection short name value to be used only if the primary or the *OfficialShortName_1* configuration value cannot be retrieved. There is no limit to the number of *OfficialShortName_X* entries. The processing software will search the configuration guide, incrementing the official short name identifier, until it cannot find an entry in the configuration guide.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.OfficialShortName_X

The *OfficialShortName_X*'s *configValues* are N_Collection_Short_Name values which are defined in the DPIS ICD. Some example values are: VIIRS-M1-FSDR, VIIRS-Prev-Ice-Age-GridIP-Rolling-Tile, etc.

3.2.1.4.1.2 SubstituteShortName_X – Name Identifier

The *SubstituteShortName_X* name identifier lists out collection short names in priority order to be used for retrieval and graceful degradation during substitute processing. Substitute processing is controlled by a command line argument flag “-substitute” which causes the processing software to look for substitute entries in the configuration guide. Substitute processing allows the system to be configured to use substitute products for inputs and to produce substitute outputs, overriding the official products initially specified. The *SubstituteShortName_1* entry will specify the primary substitute collection short name value to be used in the DMS query. *SubstituteShortName_2* entry will specify the secondary or alternate substitute collection short name value to be used only if the primary or the *SubstituteShortName_1* configuration value cannot be retrieved. There is no limit to the number of *SubstituteShortName_X* entries. The processing software will search the configuration guide, incrementing the substitute short name identifier until it cannot find an entry in the configuration guide.

If substitute processing is specified when a process is tasked and there are no substitute short name entries at all in the configuration guide, then the processing software will resort back to using the *OfficialShortName_X* entries. This flexibility is provided such that updating all the configuration guides for substitute short name entries is not mandated; it is only required in the configuration guides where substitute processing is needed. However, if there are substitute entries in the config guide and these items cannot be retrieved, the PRO software will not use official entries. It will first check for *SubstituteShortNameConversionProduct_X* entries. If one of these is present it will query for the shortname associated with that entry. If no *SubstituteShortNameConversionProduct_X* entries exist it will fail the retrieval for the item. A substitute product may be produced, in addition to an official product or in lieu of an official product.

HARDCOPY UNCONTROLLED

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>. **SubstituteShortName_X**

The *SubstituteShortName_X*'s *configValues* are N_Collection_Short_Name values which are defined in the DPIS ICD. Some example values are: VIIRS-M1-FSDR-SUB, VIIRS-Prev-Ice-Age-GridIP-Rolling-Tile-SUB, etc.

3.2.1.4.1.3 OfficialAlgBranch_X – Name Identifier

The *OfficialAlgBranch_X* name identifier lists out algorithm branches to match the OfficialShortName_X, in priority order, to be used for informing common code that a new sized product or product type is occurring; where the X represents a number. For instance, the *OfficialAlgBranch_1* entry will specify the primary algorithm branch value to be used. If not specified in the configuration guide the PRO code will assume that the algorithm branch is the "DEFAULT" branch. *OfficialAlgBranch_2* entry will specify the secondary or alternate algorithm branch value to be used only if the primary or the *OfficialShortName_1* product cannot be retrieved. There is no limit to the number of *OfficialAlgBranch_X* entries. The processing software will search the configuration guide, incrementing the algorithm branch identifier, until it cannot find an entry in the configuration guide.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>. **OfficialAlgBranch_X**

The *OfficialAlgBranch_X*'s *configValues* are algorithm branch values which are defined in ProCmnConfigGuideNames.cpp. Some example values are: DEFAULT, NCEP, CLIMO, etc.

3.2.1.4.1.4 SubstituteAlgBranch_X – Name Identifier

The *SubstituteAlgBranch_X* name identifier lists out algorithm branches in priority order to be used for informing common code that a new sized product or product type is occurring during substitute processing. Substitute processing is controlled by a command line argument flag "-substitute" which causes the processing software to look for substitute entries in the configuration guide. The *SubstituteAlgBranch_1* entry will specify the primary substitute algorithm branch value to be used. If not specified the algorithm branch will be assumed to be "DEFAULT". *SubstituteAlgBranch_2* entry will specify the secondary or alternate algorithm branch value to be used only if the primary or the *SubstituteShortName_1* configuration product cannot be retrieved. There is no limit to the number of *SubstituteAlgBranch_X* entries. The processing software will search the configuration guide, incrementing the substitute algorithm branch identifier in conjunction with the *SubstituteShortName_X* until it cannot find an entry that can be retrieved from DMS.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>. **SubstituteAlgBranch_X**

The *SubstituteAlgBranch_X*'s *configValues* are N_Collection_Short_Name values which are defined in ProCmnConfigGuideNames.cpp. Some example values are: DEFAULT, NCEP, CLIMO, etc.

3.2.1.4.1.5 SubstituteShortNameConversionProduct_X – Name Identifier

The *SubstituteShortNameConversionProduct_X* name identifier provides collection short names in priority order to be used during substitute processing when the product associated with the *SubstituteShortName_X* identifier is not found in DMS. The products associated with this identifier are the scaled version of the *SubstituteShortName_X* product. If the *SubstituteShortName_X* entry cannot be found in DMS, there is a check for a *SubstituteShortNameConversionProduct_X* identifier. If one is found, entries associated with it will be used in the DMS query. If a product is found, it will be used in processing after the data is unscaled.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.

SubstituteShortNameConversionProduct_X

The *SubstituteShortNameConversionProduct_X*'s *configValues* are N_Collection_Short_Name values which are defined in the DPIS ICD. Some

example values are: VIIRS-M1-FSDR-SUB, VIIRS-Prev-Ice-Age-GridIP-Rolling-Tile-SUB, etc.

3.2.1.4.1.6 Mode – Name Identifier

The *Mode* name identifier is used to configure a specific input as optional, required, or that no automatic setup for the item is to be performed. Inputs which are specified as required will cause program termination if all alternates (graceful degradation) have been exhausted for the input and nothing can be retrieved from DMS. An item marked optional will not cause program termination on failure to retrieve but it will be up to the derived algorithm to determine that the item was not retrieved. An input item marked to bypass automatic setup (NOAUTOSSETUP) will result in auto-code generation of a structure of pointers for the item, however no attempt is made to retrieve, nor associate the pointers to, any input data for the groupname. An input marked to bypass auto-setup will not be dynamically added to the runtime map of algorithm inputs and therefore no corresponding data item object is created; the auto-generated structure pointers will be zero (NULL) in this case. The pointers can be tested by the derived algorithm to differentiate such an input from one that exists or is a shell. This feature is useful in situations where the algorithm is required to be flexible enough to process a dynamic subset of inputs, determined merely by changing the configuration guide, without having to modify the code (due to the way auto-code generation works the algorithm must still be fully recompiled). The GTM Imagery M-Channel (M-Band) algorithm makes use of this feature to determine which six (of sixteen) Moderate Resolution FSDR inputs should be processed (refer to Appendix G for example).

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.**Mode**

Valid configuration values for *Mode* are "Required", "Optional", and "NOAUTOSSETUP". If the *Mode* identifier is not found, the default is "Required".

3.2.1.4.1.7 Wait – Name Identifier

The *Wait* name identifier is used to configure a specific input to wait the time specified in the <Algorithm Name>.Duration.Time (see Sections 3.2.1.2 and 3.2.1.3.1) before failing on retrieving an input. The PRO software will look for this entry in the configuration guide and set the DMS query timeout according to the duration time (seconds) specified.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.Wait

Valid configuration values for *Wait* are “True” and “False”. The default value is “False”, ‘do not wait any time’ during the query for an input if no entry is found in a configuration guide.

3.2.1.4.1.8 Crossgranule_Scans – Name Identifier

The *Crossgranule_Scans* name identifier is used to specify the required number of symmetric cross-granule scans for a granule-based input. This is used by PRO software to automatically obtain these additional scans of data by retrieving the input from DMS for the appropriate number of previous and subsequent granules. The PRO software will look for this entry in the configuration guide only if the derived algorithm does not implement its own `setupDataItems()` member function. If the *Crossgranule_Scans* entry is not specified, no data from cross-granule scans is required for this input and only the input for the tasked granule will be retrieved from DMS.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.Crossgranule_Scans

The *Crossgranule_Scans* name identifier is currently configured with a numerical string value denoting the required number of symmetric cross-granule scans. For example, a value of 2 would indicate 4 additional scans of data (2 before and 2 after the tasked granule) are required. Given a 12-scan granule, the item would be retrieved from 1 previous and 1 subsequent granule.

3.2.1.4.1.9 DataEndianType – Name Identifier

The *DataEndianType* name identifier is used to specify the data endianness of an algorithm input. This has no effect in PRO software. However, when an algorithm is run using the Algorithm Development Library (ADL), the ADL framework checks the *DataEndianType* value if it is running in “no metadata” mode and performs data endian conversion to the native endianness of the current platform. Note that the *DataEndianType* value is only checked for inputs when ADL is running in no metadata mode and not when running in the normal “with metadata” mode. In with metadata mode, the *DataEndianType* value is ignored for inputs and the “ByteOrder” metadata element in an input’s .asc file is checked instead.

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>.DataEndianType

Valid configuration values for *DataEndianType* for an input are “Big” or “Little”. The following rules are used for converting the endianness of an input item's data:

HARDCOPY UNCONTROLLED

- When no *DataEndianType* config entry exists, no conversion is performed.
- When the *DataEndianType* config entry is the same endian-type as the platform, no conversion is performed.
- When the *DataEndianType* config entry is the opposite endian-type of the platform, the conversion is performed.

For example, if an algorithm was executed within the ADL on a big endian platform and the *DataEndianType* for a given input was “Little”, that input would be converted to big endian format by byte-swapping the data prior to the processing stage of the algorithm.

3.2.1.5 Outputs - Group Identifier

The *Outputs* group identifier contains the list of outputs to be produced by the algorithm and maps string names to values for the metadata identifier, *N_Collection_Short_Name*, for producing outputs. An algorithm may not produce all outputs listed, but any output produced will have a corresponding entry in this section.

<Algorithm Name>.Outputs

3.2.1.5.1 ProCmnShortNameValue – Group Identifier

The *ProCmnShortNameValue* group identifier value is used to map to an actual collection short name value to be used in the creation of an item in DMS. There are multiple entries of *ProCmnShortNameValues* under the *Outputs* group identifier and *ProCmnShortNameValue* itself is also a group identifier.

<Algorithm Name>.Outputs.ProCmnShortNameValue

3.2.1.5.1.1 OfficialShortName – Name Identifier

The *OfficialShortName* name identifier specifies the collection short name to be used for creating the official output in DMS.

<Algorithm Name>.Outputs.<ProCmnShortNameValue>.OfficialShortName

The *OfficialShortName*'s configValues are *N_Collection_Short_Name* values which are defined in the DPIS ICD. Some example values are: VIIRS-M1-FSDR, VIIRS-Prev-Ice-Age-GridIP-Gran, etc

3.2.1.5.1.2 SubstituteShortName – Name Identifier

The *SubstituteShortName* name identifier specifies the collection short name to be used for creating substitute output item in DMS.

If substitute processing is specified when a process is tasked and there are no substitute short name output entries at all in the configuration guide, the processing software will resort back to using the *OfficialShortName* entry. This flexibility is provided such that updating all the configuration guides for substitute short name entries is not mandated but only the configuration guides where substitute processing is needed. A substitute product may be produced in addition to an official product.

<Algorithm Name>.Outputs.<ProCmnShortNameValue>. **SubstituteShortName**

The *SubstituteShortName*'s *configValues* are *N_Collection_Short_Name* values which are defined in the DPIS ICD. Some example values are: VIIRS-M1-FSDR-SUB, VIIRS-Prev-Ice-Age-GridIP-Rolling-Tile-SUB, etc.

3.2.1.5.1.3 Mode

The *Mode* name identifier, for output items, is only used when necessary to configure a given output such that automatic setup of it's associated output data item and buffer is bypassed. The *Mode* identifier is otherwise ignored for outputs. An output item marked to bypass automatic setup (NOAUTOSSETUP) will result in auto-code generation of a structure of pointers for the item, however no attempt is made to setup an output buffer, nor associate the pointers to, any output data for the groupname. Further, the output item will not be dynamically added to the runtime map of algorithm outputs and therefore no corresponding data item object is created; the auto-generated structure pointers will be zero (NULL) in this case. This feature is useful in situations where the algorithm will determine at run-time whether or not to produce the output. Such an algorithm, after determining that the output is to be produced, will manually create the data output item and add it to the map of DMS items for output. The VIIRS SDR ATMSRemap algorithm makes use of this feature for CrIS-SDR-RGEO as it's both an algorithm optional-input and output; if the CrIS SDR RGEO input doesn't exist, then the algorithm creates, uses, and outputs it, otherwise the input is used and there's is no need to re-create it as an output (refer to Appendix H for example).

<Algorithm Name>.Inputs.[Spacecraft.]<ProCmnShortNameValue>. **Mode**

The only valid, used, configuration value for *Mode* for outputs is "NOAUTOSSETUP".

3.2.1.5.1.4 OutputType – Name Identifier

The *OutputType* name identifier is used to specify the output type of a given output. An output can be specified to be put into DMS or to heap memory. The PRO software will look at the configuration guide and create the correct output type for the output based on this entry.

<Algorithm Name>.Outputs.<ProCmnShortNameValue>. **OutputType**

Valid configuration values for *OutputType* are "DMS" and "HEAP". The default behavior is to output the item to DMS, if no configuration entry is specified for a given output in the

configuration guide. HEAP output is specified here to support the Segment Design for Granule Ancillary output.

3.2.1.5.1.5 *DataEndianType* – Name Identifier

The *DataEndianType* name identifier is used to specify the desired data endianness of an algorithm output. This has no effect in PRO software. However, when an algorithm is run using the Algorithm Development Library (ADL), the ADL framework checks the *DataEndianType* value and performs data endian conversion to the desired endian format(s) before the algorithm finishes. Note that unlike *DataEndianType* for inputs, the *DataEndianType* value is checked when ADL is running in both “no metadata” and “with metadata” modes.

<Algorithm Name>.Outputs.<ProCmnShortNameValue>.***DataEndianType***

Valid configuration values for *DataEndianType* for an output are “Big”, “Little”, or “Both”. If no *DataEndianType* is specified, no endianness conversion is performed. The following rules are used for converting the endianness of an output’s data:

- When no *DataEndianType* config entry exists, no conversion is performed.
- When the *DataEndianType* config entry is the same endian-type as the platform, no conversion is performed.
- When the *DataEndianType* config entry is the opposite endian-type of the platform, the conversion is performed before the item is written. The output is written with a “.LE” or “.BE” (for little endian and big endian respectively) appended to the file name. Note that “.LE” and “.BE” are only appended to file names of outputs with a different endianness than the platform.
- When the *DataEndianType* config entry is “Both”, both a little endian and a big endian version of the output are written. The output of the opposite endian type will have a “.LE” or “.BE” (for little endian and big endian respectively) appended to the file name. The output of the native endian type will not have a “.LE” or “.BE” appended to the file name.

For example, if an algorithm was executed within the ADL on a big endian platform and the *DataEndianType* for a given output was “Little”, that output would be converted to little endian format by byte-swapping the data prior to output. The file name of the output would end in “.LE”.

As another example, if an algorithm was executed within the ADL on a little endian platform and the *DataEndianType* for a given output was “Both”, that output would be written as both little endian and big endian. The big endian output would be converted by byte-swapping the data prior to output. The little endian output would not have a “.LE” extension appended while the big endian output would have a “.BE” appended to the file name.

3.2.1.5.2 N_LEOA_Flag – Metadata Set Flag

The *N_LEOA_Flag* name identifier is used to set the *N_LEOA_Flag* metadata value for the product associated to this algorithm and output shortname/groupname. The valid values are “0” and “1.” When this flag is not given, the default is set to zero. Any invalid values supplied result in the value being set to zero.

<Algorithm Name>.Outputs.<ProCmnShortNameValue>. **N_LEOA_Flag**

3.2.2 Controller Algorithms

Configuration entries common across controller algorithms are listed in this section. These entries will only be found in PRO configuration guides for a controller algorithm.

The following subsections will describe configuration values which are specific to a controller configuration guide. An example controller configuration guide is shown in Appendix B.

3.2.2.1 SubAlgorithms - Group Identifier

The *SubAlgorithms* group identifier contains the configuration groups and names listed under this subsection. The following *SubAlgorithms* configuration items are required of all controller algorithms.

<Algorithm Name>.**SubAlgorithms**

3.2.2.1.1 Algorithm_X – Group Identifier

The *Algorithm_X* group identifier value is used to specify the algorithms which will be controlled by the controller algorithm; where X is a number starting with 1 and repeating as needed to specify all of the algorithms being controlled and each individual algorithm's configuration values. The PRO software will read algorithm entries beginning with *Algorithm_1*, incrementing the number each time until it exhausts the entries in the configuration guide. The algorithms will be executed in sequential order based on their numerical order in the configuration guide, i.e., *Algorithm_1*, *Algorithm_2*, *Algorithm_3*, etc.

<Algorithm Name>.SubAlgorithms.**Algorithm_X**

3.2.2.1.1.1 AlgorithmName – Name Identifier

The *AlgorithmName* name identifier value is used to specify the algorithm name which will be controlled by the controller algorithm. The naming convention to be followed is that the algorithm name should be the same as the derived algorithm class. For instance, ProSdrViirs would be the algorithm name for the SDR VIIRS derived algorithm.

<Algorithm Name>.SubAlgorithms.Algorithm_X.<**AlgorithmName**>

The *AlgorithmName* configuration values will be defined in ProCmnAlgorithmNames.h and ProCmnAlgorithmNames.cpp.

3.2.2.1.1.2 Params – Group Identifier

The *Params* group identifier is used to specify each controlled algorithm's parameters, such that an algorithm object can be instantiated by the controller algorithm. As a general rule, there should be one *Param_X* (See Section 3.2.2.1.1.2.1) entry for each argument passed to the constructor of the algorithm.

<Algorithm Name>.SubAlgorithms.Algorithm_X.Params

3.2.2.1.1.2.1Param_X – Name Identifier

The *Param_X* name identifier is used to specify each individual parameter for a given algorithm. The X is a number beginning with 1 and increasing by one for each parameter specified for an algorithm. These parameters are repeating as necessary to define all parameters to instantiate a controlled algorithm. Also, certain parameter values can be used to tell the PRO software to look at other configuration guides in order to build the correct parameter value. For instance, the GridToGran controller algorithm will use a special BUILD_BITMASK value to tell the PRO software to look into the controlled GridToGran configuration guides such that an algorithm bit mask can be generated on the fly for instantiating the GridToGran controller algorithm.

<Algorithm Name>.SubAlgorithms.Algorithm_X.Params.Param_X

All values are string values and each parameter will be interpreted by the specific algorithm factory functionality which is used to instantiate the algorithm. Example values to be found in a configuration guide are: DMAPICLIENT, IPQM, BUILD_BITMASK; where DMAPICLIENT tells the PRO software to use the DMS client from the controller algorithm, IPQM denotes the stages to be executed for the algorithm, and BUILD_BITMASK tells the PRO software to build the algorithm bit mask from the configuration guides. All new *Param_X* configuration values need to be coordinated with the algorithm factory software so that the correct interpretation of the parameter is done. Adding new parameters will take corresponding code changes to the algorithm factory to interpret the parameter correctly.

The current parameters are defined as follows:

- DMAPICLIENT – Instructs the algorithm factory to make use of the DMS client pointer which is passed into the factory.
- IPQM – Instructs the algorithm factory to create a stage mask that includes input, processing, quality notification, and metadata.
- IPQMO - Instructs the algorithm factory to create a stage mask that includes input, processing, quality notification, metadata, and output.

New parameters should be documented above as added to specific algorithm factory functionality.

3.2.3 GridToGran Algorithm

The specific GridToGran configuration guides contain these additional configuration values which are placed under the general *ConfigValues* group identifier; see Section 3.2.1.2.

3.2.3.1 EnvironmentRunName – Name Identifier

The *EnvironmentRunNameOfficial* name identifier value is used to specify the algorithm in which this subalgorithm will be executed out of. A particular subalgorithm will run in only one environment at a time. Potential execution environments can specify running from within an SDR controller, an EDR controller, or even as a stand-alone algorithm. *EnvironmentRunNameSub* is the name identifier value used to specify the algorithm which will execute the subalgorithm during substitute processing. Substitute processing is controlled by a command line argument flag “-substitute” which causes the processing software to look for substitute entries in the configuration guide.

<Algorithm Name>.ConfigValues.**EnvironmentRunNameOfficial**

<Algorithm Name>.ConfigValues.**EnvironmentRunNameSub**

The *EnvironmentRunName* configuration values will be defined in ProCmnAlgorithmNames.h/ProCmnAlgorithmNames.cpp. An example value would be ProSdrViirsController. These entries are read by the PRO software to determine the correct algorithm bit mask to be used to instantiate the GridToGran controller algorithm. When executing out of the ProSdrController, the PRO software will look for GridToGran algorithms which have the *EnvironmentRunName* set to ProSdrViirsController.

3.2.3.2 Constellation – Name Identifier

The *Constellation* name identifier value is used to specify the platform (individual spacecraft or constellation) to be used for this granulated tile input item. *Constellation* is only valid for tile input items.

<Algorithm Name>.ConfigValues.**Constellation**

The valid values for *Constellation* are a valid spacecraft identifier (“N01”, “N02”) or “CONST”.

3.2.4 GranToGrid Algorithm

The GranToGrid algorithm configuration guides contain additional configuration values which are placed under the general *ConfigValues* group identifier, see Section 3.2.1.2. Since the GranToGrid processes are persistent processes, the configuration guides are read only once up front and that the processes have to be terminated and restarted for any configuration changes on GIP exclusion.

3.2.4.1 AllowGracefulDegradationFlag – Name Identifier

The *AllowGracefulDegradationFlag* name identifier is used to specify whether or not to validate whether graceful degradation was used to determine the data values. When the value is set to “False”, the algorithm will go out and check the metadata to determine whether or not graceful degradation is present in the data, if the value in metadata is “Yes”, then the GIP Exclusion will be utilized, and the tiles are not updated. A value of “True” skips this check, since you’re configuring to allow graceful degradation in the data, and updates the tiles as in normal processing.

<Algorithm Name>.ConfigValues.**AllowGracefulDegradationFlag**

3.2.4.2 AllowSpacecraftManeuversFlag – Name Identifier

The *AllowSpacecraftManeuversFlag* name identifier is used to specify whether or not to validate whether a spacecraft maneuver was used to determine the data values. When the value is set to “False”, the algorithm will go out and check the metadata to determine whether or not spacecraft maneuvers has anything other than a value of “Normal Operations”, then the GIP Exclusion will be utilized, which mean the tiles will not be updated. A value of “True” skips this check, since you’re configuring to allow spacecraft maneuvers, and updates the tiles as in normal processing.

<Algorithm Name>.ConfigValues.**AllowSpacecraftManeuversFlag**

3.2.4.3 ProGipPseudo – Name Identifier

The *ProGipPseudo* name identifier is used to store the common short name used when logging status messages.

<Algorithm Name>.ConfigValues.**ProGipPseudo**

3.2.4.4 Constellation – Name Identifier

The *Constellation* name identifier value is used to specify the individual spacecraft or constellation to be used for this granulated tile input item

<Algorithm Name>.ConfigValues.**Constellation**

3.2.5 Granulate Ancillary Algorithm

The specific Granulate Ancillary configuration guides contain these additional configurations values which are placed under the general *ConfigValues* group identifier, see Section 3.2.1.2.

3.2.5.1 EnvironmentRunName – Name Identifier

The *EnvironmentRunNameOfficial* name identifier value is used to specify the algorithm which will execute the Gran Anc algorithm. A particular Gran Anc algorithm will run in only one environment at a time. The potential execution environments are the SDR controller algorithm, EDR controller algorithm, or as a stand-alone algorithm. *EnvironmentRunNameSub* is the name identifier value used to specify the algorithm which will execute the Gran Anc algorithm during substitute processing. Substitute processing is controlled by a command line argument flag “-substitute” which causes the processing software to look for substitute entries in the configuration guide.

<Algorithm Name>.ConfigValues.**EnvironmentRunNameOfficial**

<Algorithm Name>.ConfigValues.**EnvironmentRunNameSub**

3.2.6 Granulate Ancillary Controller

Configuration entries specific to Granulate Ancillary (Gran Anc) are listed in this section. These entries will only be found in PRO configuration guides for a Gran Anc controller algorithm. Each sensor-specific algorithm requiring Gran Anc data will require a Gran Anc controller.

An example controller configuration guide is shown in Appendix C

3.2.7 WCalc Algorithm

The Weight Calculation (WCalc) algorithm configuration guides contain additional configuration values which are placed under the general *ConfigValues* group identifier, see Section 3.2.1.2.

3.2.7.1 NumberOfDetectors – Name Identifier

The *NumberOfDetectors* name identifier is used to specify the number of detectors that the algorithm should use when determining if a row is on a scan boundary. The WCalc algorithm performs slightly different calculations depending on if a row is on a scan boundary or not. This configuration value is placed in the WCalc algorithm configuration guide rather than a variable defined in the code in order to give the WCalc algorithm the ability to change the number of detectors the algorithm uses during processing without the need to recompile the code.

3.2.7.2 AggregationZones – Group Identifier

The *AggregationZones* group identifier contains a list of aggregation zones which should be applied when the data is being processed by the WCalc algorithm.

3.2.7.2.1 AggZone_X – Group Identifier

The *AggZone_X* group identifier contains configuration information for a given aggregation zone. The first aggregation zone will be *AggZone_1*, the second aggregation zone will be *AggZone_2*, and so on.

3.2.7.2.1.1 StartPixelColumn – Name Identifier

The *StartPixelColumn* name identifier specifies an integer number which is the column number of the data that the current aggregation zone starts on.

<AlgorithmName>.ConfigValues.AggregationZones.AggZone_<X>.**StartPixelColumn**

3.2.7.2.1.2 NumPixelsInZone – Name Identifier

The *NumPixelsInZone* name identifier specifies the number of pixels in the current aggregation zone.

<AlgorithmName>.ConfigValues.AggregationZones.AggZone_<X>.**NumPixelsInZone**

3.2.7.2.1.3 SmearFactor – Name Identifier

The *SmearFactor* name identifier specifies the smear factor which was applied to the current aggregation zone.

<AlgorithmName>.ConfigValues.AggregationZones.AggZone_<X>.**SmearFactor**

APPENDIX A – EXAMPLE OF GENERAL ALGORITHM CONFIGURATION GUIDE

<i>Configuration Guide Search Strings for General Algorithms</i>	<i>Values</i>
<i>AlgorithmName.Algorithm.Version</i>	<i>String value ("1.0.000.001")</i>
<i>AlgorithmName.Inputs.ProCmnShortNameValue.OfficialShortName_X</i>	<i>ProCmnShortNames.h</i>
<i>AlgorithmName.Inputs.ProCmnShortNameValue.SubstituteShortName_X</i>	<i>ProCmnShortNames.h</i>
<i>AlgorithmName.Inputs.ProCmnShortNameValue.Mode</i>	<i>Required or Optional</i>
<i>AlgorithmName.Outputs.ProCmnShortNameValue.OfficialShortName</i>	<i>ProCmnShortNames.h</i>
<i>AlgorithmName.Outputs.ProCmnShortNameValue.SubstituteShortName</i>	<i>ProCmnShortNames.h</i>
<i>AlgorithmName.Outputs.ProCmnShortNameValue.OutputType</i>	<i>DMS or HEAP</i>

```
<?xml version="1.0" encoding="UTF-8"?>
<CFG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CFGSchema.xsd">
```

```
<!-- -->
<!-- This group contains VIIRS Vegetation Index config parameters -->
<!-- -->
```

```
<group name="ProEdrViirsVI">
```

```
  <group name="Algorithm">
    <config>
      <name>Version</name>
      <configValue>1.0.001.001</configValue>
    </config>
  </group>
```

```
<!-- -->
<!-- INPUTS -->
<!-- -->
```

```
<group name="Inputs">
```

```
  <group name="Geolocation_Img_TC">
    <config>
      <name>OfficialShortName_1</name>
      <configValue>VIIRS-IMG-RGEO-TC</configValue>
    </config>
    <config>
      <name>OfficialShortNameConversionProduct_1</name>
      <configValue>VIIRS-IMG-GEO-TC</configValue>
    </config>
  </group>
```

```
  <group name="Image_Band01">
    <config>
      <name>OfficialShortName_1</name>
      <configValue>VIIRS-I1-FSDR</configValue>
    </config>
    <config>
      <name>OfficialShortNameConversionProduct_1</name>
      <configValue>VIIRS-I1-SDR</configValue>
    </config>
  </group>
```

```
  <group name="Image_Band02">
    <config>
      <name>OfficialShortName_1</name>
      <configValue>VIIRS-I2-FSDR</configValue>
    </config>
    <config>
      <name>OfficialShortNameConversionProduct_1</name>
```

HARDCOPY UNCONTROLLED

```

        <configValue>VIIRS-I2-SDR</configValue>
    </config>
</group>

<group name="SurfReflect_IP">
    <config>
        <name>OfficialShortName_1</name>
        <configValue>VIIRS-Surf-Refl-IP</configValue>
    </config>
    <config>
        <name>SubstituteShortName_1</name>
        <configValue>VIIRS-Surf-Refl-IP-SUB</configValue>
    </config>
</group>

<group name="VI_Pro_Coeffs">
    <config>
        <name>OfficialShortName_1</name>
        <configValue>VIIRS-VI-EDR-AC-Int</configValue>
    </config>
</group>

<group name="VI_DQTT">
    <config>
        <name>Mode</name>
        <configValue>Optional</configValue>
    </config>
    <config>
        <name>OfficialShortName_1</name>
        <configValue>VIIRS-VI-EDR-DQTT-Int</configValue>
    </config>
</group>
</group>

<!--      -->
<!--  OUTPUTS  -->
<!--      -->

<group name="Outputs">
    <group name="VI_EDR_DQN">
        <config>
            <name>OfficialShortName</name>
            <configValue>VIIRS-VI-EDR-DQN</configValue>
        </config>
        <config>
            <name>SubstituteShortName</name>
            <configValue>VIIRS-VI-EDR-DQN-SUB</configValue>
        </config>
    </group>

    <group name="VegIndex_EDR">
        <config>
            <name>OfficialShortName</name>
            <configValue>VIIRS-VI-FEDR</configValue>
        </config>
        <config>
            <name>OutputType</name>
            <configValue>HEAP</configValue>
        </config>
        <config>
            <name>OfficialScaledShortName</name>
            <configValue>VIIRS-VI-EDR</configValue>
        </config>
        <config>
            <name>SubstituteShortName</name>
            <configValue>VIIRS-VI-FEDR-SUB</configValue>
        </config>
    </group>
</group>

```

HARDCOPY UNCONTROLLED

```
</config>
<config>
  <name>SubstituteScaledShortName</name>
  <configValue>VIIRS-VI-EDR-SUB</configValue>
</config>
</group>

</group>

</group>

</CFG>
```

HARDCOPY UNCONTROLLED

APPENDIX B – EXAMPLE CONTROLLER ALGORITHM CONFIGURATION GUIDE*Configuration Guide Search strings additions for Controller Algorithms**Values**AlgorithmName.SubAlgorithms.Algorithm_X.AlgorithmName**ProCmnAlgorithmNames.h**AlgorithmName.SubAlgorithms.Algorithm_X.Params.Param_X**DMAPICLIENT, IPQM*

```

<?xml version="1.0" encoding="UTF-8"?>
<CFG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CFGSchema.xsd">

  <group name="ProEdrViirsMasksController">

    <group name="Algorithm">

      <config>
        <name>Version</name>
        <configValue>1.0.001.001</configValue>
      </config>
    </group>

    <!-- ***** -->
    <!-- Define the SubAlgorithms - those that are part of this controller. -->
    <!-- ***** -->
    <group name="SubAlgorithms">

      <!-- ***** -->
      <!-- Gran Anc Controller -->
      <!-- ***** -->
      <group name="Algorithm_1">
        <config>
          <name>AlgorithmName</name>
          <configValue>ProAncViirsController</configValue>
        </config>

        <!-- if this is set, then all output products -->
        <!-- for this algorithm are pre-released when -->
        <!-- this subalgo completes. -->
        <!-- (Used during substitute processing -->

        <config>
          <name>PrereleaseAllWhenFinished</name>
          <configValue>true</configValue>
        </config>

        <!-- if this is set, then all pre-released -->
        <!-- output products will also generate an -->
        <!-- XDR GRAN message at time of release -->
        <!-- (versus at the end of the Controller) -->

        <config>
          <name>WFM_XDR_anyReleased</name>
          <configValue>true</configValue>
        </config>

      <group name="Params">
        <config>
          <name>Param_1</name>
          <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
          <name>Param_2</name>
          <configValue>IPQM</configValue>
        </config>
      </group>
    </group>
  </group>

```

HARDCOPY UNCONTROLLED

```
</group>

<!-- End Gran Anc Controller -->

<!-- ***** -->
<!-- Active Fires Algorithm -->
<!-- ***** -->
<group name="Algorithm_2">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProEdrViirsActiveFires</configValue>
  </config>

  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group>
<!-- End Active Fires Algorithm -->

<!-- ***** -->
<!-- VIIRS Cloud Mask IP Algorithm -->
<!-- ***** -->
<group name="Algorithm_3">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProEdrViirsCM</configValue>
  </config>

  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group>
<!-- ***** -->
<!-- VIIRS Granulated AOT Climatology Algorithm -->
<!-- ***** -->
<group name="Algorithm_4">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateAotClimatology</configValue>
  </config>

  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group>
<!-- End SubAlgorithm group -->
</group>
```

HARDCOPY UNCONTROLLED

</CFG>

HARDCOPY UNCONTROLLED

APPENDIX C – EXAMPLE GRANULATE ANCILLARY-SPECIFIC CONTROLLER ALGORITHM CONFIGURATION GUIDE

Configuration Guide Search strings additions for Controller Algorithms	Values
<i>AlgorithmName.ConfigValues.EnvironmentsRunNameX</i>	<i>AlwaysCreate</i>
<i>AlgorithmName.SubAlgorithms.Algorithm_X.Params.Param_X</i>	<i>DMAPICLIENT, IPQM</i>

```
<?xml version="1.0" encoding="UTF-8"?>
<CFG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CFGSchema.xsd">

  <group name="ProAncViirsController">
    <group name="ConfigValues">
      <config>
        <name>EnvironmentRunNameOfficial</name>
        <configValue>AlwaysCreate</configValue>
      </config>
      <config>
        <name>EnvironmentRunNameSub</name>
        <configValue>AlwaysCreate</configValue>
      </config>
    </group>
    <!-- End ConfigValues -->
    <group name="Algorithm">
      <config>
        <name>Version</name>
        <configValue>1.0.001.001</configValue>
      </config>
    </group>

    <!-- Define the SubAlgorithms - those that are part of this controller. -->
    <group name="SubAlgorithms">
      <group name="Algorithm_1">
        <config>
          <name>AlgorithmName</name>
          <configValue>ProAncViirsGranulateTerrainGeopotentialHeight</configValue>
        </config>
        <group name="Params">
          <config>
            <name>Param_1</name>
            <configValue>DMAPICLIENT</configValue>
          </config>
          <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
          </config>
        </group>
      </group>
    </group> <!-- End of ProAncViirsGranulateTerrainGeopotentialHeight Algorithm -->
    <group name="Algorithm_2">
      <config>
        <name>AlgorithmName</name>
        <configValue>ProAncViirsGranulateWindSpeed</configValue>
      </config>
      <group name="Params">
        <config>
          <name>Param_1</name>
          <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
          <name>Param_2</name>
          <configValue>IPQM</configValue>
        </config>
      </group>
    </group>
  </group>
</CFG>
```

HARDCOPY UNCONTROLLED

```
</group> <!-- End of ProAncViirsGranulateWindSpeed Algorithm -->
<group name="Algorithm_3">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateWindDirection</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateWindDirection Algorithm -->
<group name="Algorithm_4">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateOzone</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateOzone Algorithm -->
<group name="Algorithm_5">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulatePrecipWater</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulatePrecipWater Algorithm -->
<group name="Algorithm_6">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulatePresLevelTemp</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulatePresLevelTemp Algorithm -->
<group name="Algorithm_7">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateWaterVaporMixRatio</configValue>
```

HARDCOPY UNCONTROLLED

```
</config>
<group name="Params">
  <config>
    <name>Param_1</name>
    <configValue>DMAPICLIENT</configValue>
  </config>
  <config>
    <name>Param_2</name>
    <configValue>IPQM</configValue>
  </config>
</group>
</group> <!-- End of ProAncViirsGranulateWaterVaporMixRatio Algorithm -->
<group name="Algorithm_8">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateSpecSurfHumidity</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateSpecSurfHumidity Algorithm -->
<group name="Algorithm_9">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateSurfTemp</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateSurfTemp Algorithm -->
<group name="Algorithm_10">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateTropoGeopotentialHeight</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateTropoGeopotentialHeight Algorithm -->
<group name="Algorithm_11">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateSurfGeopotentialHeight</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
```

HARDCOPY UNCONTROLLED

```

        </config>
        <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
        </config>
    </group>
</group> <!-- End of ProAncViirsGranulateSurfGeopotentialHeight Algorithm -->
<group name="Algorithm_12">
    <config>
        <name>AlgorithmName</name>
        <configValue>ProAncViirsGranulateNitrateDepletion</configValue>
    </config>
    <group name="Params">
        <config>
            <name>Param_1</name>
            <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
        </config>
    </group>
</group> <!-- End of ProAncViirsGranulateNitrateDepletion Algorithm -->
<group name="Algorithm_13">
    <config>
        <name>AlgorithmName</name>
        <configValue>ProAncViirsGranulateBathymetry</configValue>
    </config>
    <group name="Params">
        <config>
            <name>Param_1</name>
            <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
        </config>
    </group>
</group> <!-- End of ProAncViirsGranulateBathymetry Algorithm -->
<group name="Algorithm_14">
    <config>
        <name>AlgorithmName</name>
        <configValue>ProAncViirsGranulateOpticalDepth</configValue>
    </config>
    <group name="Params">
        <config>
            <name>Param_1</name>
            <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
        </config>
    </group>
</group> <!-- End of ProAncViirsGranulateOpticalDepth Algorithm -->
<group name="Algorithm_15">
    <config>
        <name>AlgorithmName</name>
        <configValue>ProAncViirsGranulateSkinTemp</configValue>
    </config>
    <group name="Params">
        <config>
            <name>Param_1</name>
            <configValue>DMAPICLIENT</configValue>
        </config>
        <config>
            <name>Param_2</name>
            <configValue>IPQM</configValue>
        </config>
    </group>
</group>

```

HARDCOPY UNCONTROLLED

```

    </group>
</group> <!-- End of ProAncViirsGranulateSkinTemp Algorithm -->
<group name="Algorithm_16">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateSurfPres</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateSurfPres Algorithm -->
<group name="Algorithm_17">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateGeopotentialHeight</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateGeopotentialHeight Algorithm -->
<group name="Algorithm_18">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsPerformSurfPresCorrection</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsPerformSurfPresCorrection Algorithm -->
<group name="Algorithm_19">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateNhfSurfTemp</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateSurfTemp Algorithm -->
<group name="Algorithm_20">
  <config>
    <name>AlgorithmName</name>

```

HARDCOPY UNCONTROLLED

```

    <configValue>ProAncViirsGranulateNhfsurfPres</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateNhfsurfPres Algorithm -->
<group name="Algorithm_21">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateNhfspecsurfHumidity</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateNhfspecsurfHumidity Algorithm -->
<group name="Algorithm_22">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateNhfpresLevelTemp</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateNhfpresLevelTemp Algorithm -->
<group name="Algorithm_23">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateNhfwaterVaporMixRatio</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>
      <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
      <name>Param_2</name>
      <configValue>IPQM</configValue>
    </config>
  </group>
</group> <!-- End of ProAncViirsGranulateNhfwaterVaporMixRatio Algorithm -->
<group name="Algorithm_24">
  <config>
    <name>AlgorithmName</name>
    <configValue>ProAncViirsGranulateNhfozone</configValue>
  </config>
  <group name="Params">
    <config>
      <name>Param_1</name>

```

HARDCOPY UNCONTROLLED

```
        <configValue>DMAPICLIENT</configValue>
    </config>
    <config>
        <name>Param_2</name>
        <configValue>IPQM</configValue>
    </config>
</group>
</group> <!-- End of ProAncViirsGranulateNhf0zone Algorithm -->
</group> <!-- End of SubAlgorithms -->
</group> <!-- ProAncViirsController -->

</CFG>
```

HARDCOPY UNCONTROLLED

APPENDIX D – EXAMPLE GRAN ANC ALGORITHM CONFIGURATION GUIDE

Configuration Guide Search strings additions for Gran Anc Algorithms	Values
AlgorithmName.ConfigValues.EnvironmentRunName	ProCmnAlgorithmNames.h

```

<?xml version="1.0" encoding="UTF-8"?>
<CFG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CFGSchema.xsd">

  <!--
  group name: ProAncViirsGranulateWindSpeed
  desc: Config guide for VIIRS Granulated Wind Speed
  -->
  <group name="ProAncViirsGranulateWindSpeed">
    <group name="ConfigValues">
      <config>
        <name>EnvironmentRunNameOfficial</name>
        <configValue>ProSdrViirsController</configValue>
      </config>
      <config>
        <name>EnvironmentRunNameSub</name>
        <configValue>ProEdrViirsMasksController</configValue>
      </config>
    </group>

    <group name="Algorithm">
      <config>
        <name>Version</name>
        <configValue>1.0.001.001</configValue>
      </config>
    </group>

    <!-- Input items needed -->
    <group name="Inputs">
      <group name="Geolocation_Mod_TC">
        <config>
          <name>DataEndianType</name>
          <configValue>Big</configValue>
        </config>
        <config>
          <name>OfficialShortName_1</name>
          <configValue>VIIRS-MOD-RGEO-TC</configValue>
        </config>
        <config>
          <name>OfficialShortNameConversionProduct_1</name>
          <configValue>VIIRS-MOD-GEO-TC</configValue>
        </config>
      </group>
    </group>
  </group>

```

HARDCOPY UNCONTROLLED

Appendix E – Example wcalc algorithm configuration guide

Configuration Guide Search strings additions WCalc Algorithms	Values
AlgorithmName.Algorithm.Version	1.0.000.001
AlgorithmName.ConfigValues.EnvironmentRunName	ProSdrViirsController
AlgorithmName.ConfigValues.NumberOfDetectors	16
AlgorithmName.ConfigValues.AggregationZones.AggZone_1.StartPixelColumn	0
AlgorithmName.ConfigValues.AggregationZones.AggZone_1.NumPixelsInZone	640
AlgorithmName.ConfigValues.AggregationZones.AggZone_1.SmearFactor	1.0
AlgorithmName.ConfigValues.AggregationZones.AggZone_2.StartPixelColumn	640
AlgorithmName.ConfigValues.AggregationZones.AggZone_2.NumPixelsInZone	368
AlgorithmName.ConfigValues.AggregationZones.AggZone_2.SmearFactor	0.5
AlgorithmName.ConfigValues.AggregationZones.AggZone_3.StartPixelColumn	1008
AlgorithmName.ConfigValues.AggregationZones.AggZone_3.NumPixelsInZone	592
AlgorithmName.ConfigValues.AggregationZones.AggZone_3.SmearFactor	0.3333333
AlgorithmName.ConfigValues.AggregationZones.AggZone_4.StartPixelColumn	1600
AlgorithmName.ConfigValues.AggregationZones.AggZone_4.NumPixelsInZone	592
AlgorithmName.ConfigValues.AggregationZones.AggZone_4.SmearFactor	0.3333333
AlgorithmName.ConfigValues.AggregationZones.AggZone_5.StartPixelColumn	2192
AlgorithmName.ConfigValues.AggregationZones.AggZone_5.NumPixelsInZone	368
AlgorithmName.ConfigValues.AggregationZones.AggZone_5.SmearFactor	0.5
AlgorithmName.ConfigValues.AggregationZones.AggZone_6.StartPixelColumn	2560
AlgorithmName.ConfigValues.AggregationZones.AggZone_6.NumPixelsInZone	640
AlgorithmName.ConfigValues.AggregationZones.AggZone_6.SmearFactor	1.0
AlgorithmName.Inputs.Geolocation_Mod_TC.OfficialShortName_1	VIIRS-MOD-RGEO-TC
AlgorithmName.Outputs.AreaWeight_Mod_Gran.OfficialShortName	VIIRS-Grid-To-Gran-GridIP-AW-SWATH-Mod-IP

```
<?xml version="1.0" encoding="UTF-8"?>
<CFG xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CFGSchema.xsd">

  <group name="ProGipViirsWCalcGran">

    <group name="Algorithm">
      <config>
        <name>Version</name>
        <configValue>1.0.001.001</configValue>
      </config>
    </group>

    <!-- Configuration Values such as aggregation zones -->
    <group name="ConfigValues">

      <config>
        <name>EnvironmentRunNameOfficial</name>
        <configValue>ProSdrViirsController</configValue>
      </config>
      <config>
        <name>EnvironmentRunNameSub</name>
        <configValue>ProSdrViirsController</configValue>
      </config>

      <!-- For MODIS -->
      <!--
      <config>
        <name>NumberOfDetectors</name>
        <configValue>10</configValue>
      </config>
    -->
  </group>
</CFG>
```

HARDCOPY UNCONTROLLED

```
<!-- For VIIRS -->
<config>
  <name>NumberOfDetectors</name>
  <configValue>16</configValue>
</config>

<group name="AggregationZones">
  <!-- For VIIRS -->
  <group name="AggZone_1">
    <config>
      <name>StartPixelColumn</name>
      <configValue>0</configValue>
    </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>640</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>1.0</configValue>
    </config>
  </group>
  <group name="AggZone_2">
    <config>
      <name>StartPixelColumn</name>
      <configValue>640</configValue>
    </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>368</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>0.5</configValue>
    </config>
  </group>
  <group name="AggZone_3">
    <config>
      <name>StartPixelColumn</name>
      <configValue>1008</configValue>
    </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>592</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>0.333333</configValue>
    </config>
  </group>
  <group name="AggZone_4">
    <config>
      <name>StartPixelColumn</name>
      <configValue>1600</configValue>
    </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>592</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>0.333333</configValue>
    </config>
  </group>
  <group name="AggZone_5">
    <config>
      <name>StartPixelColumn</name>
```

HARDCOPY UNCONTROLLED

```

        <configValue>2192</configValue>
      </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>368</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>0.5</configValue>
    </config>
  </group>
  <group name="AggZone_6">
    <config>
      <name>StartPixelColumn</name>
      <configValue>2560</configValue>
    </config>
    <config>
      <name>NumPixelsInZone</name>
      <configValue>640</configValue>
    </config>
    <config>
      <name>SmearFactor</name>
      <configValue>1.0</configValue>
    </config>
  </group>
</group>
<!-- End ConfigValues -->

<!-- Start Inputs -->
<group name="Inputs">
  <group name="Geolocation_Mod_TC">
    <config>
      <name>DataEndianType</name>
      <configValue>Big</configValue>
    </config>
    <config>
      <name>OfficialShortName_1</name>
      <configValue>VIIRS-MOD-RGEO-TC</configValue>
    </config>
    <config>
      <name>OfficialShortNameConversionProduct_1</name>
      <configValue>VIIRS-MOD-GEO-TC</configValue>
    </config>
  </group>
</group>
<!-- End Inputs -->

<!-- Start Outputs -->
<group name="Outputs">
  <group name="AreaWeight_Mod_Gran">
    <config>
      <name>DataEndianType</name>
      <configValue>Both</configValue>
    </config>
    <config>
      <name>OfficialShortName</name>
      <configValue>VIIRS-Grid-To-Gran-GridIP-AW-SWATH-Mod-IP</configValue>
    </config>
  </group>
</group>
<!-- End Outputs -->

</group>
</CFG>

```

HARDCOPY UNCONTROLLED

Appendix G – Example Using NOAUTOSSETUP Mode (from GTM Imagery MBand Configuration Guide)

Note: only first two of sixteen Moderate Band inputs shown.

```
<group name="Moderate_Band01">
  <config>
    <name>DataEndianType</name>
    <configValue>Big</configValue>
  </config>
  <config>
    <!-- algorithm's cross-granule scan requirements exclusive -->
    <!-- of any extended granule requirements -->
    <name>Crossgranule_Scans</name>
    <configValue>2</configValue>
  </config>
  <config>
    <name>Mode</name>
    <configValue>Required</configValue>
  </config>
  <config>
    <name>Wait</name>
    <configValue>True</configValue>
  </config>
  <config>
    <name>OfficialShortName_1</name>
    <configValue>VIIRS-M1-FSDR</configValue>
  </config>
  <config>
    <name>OfficialShortNameConversionProduct_1</name>
    <configValue>VIIRS-M1-SDR</configValue>
  </config>
</group>

<group name="Moderate_Band02">
  <config>
    <name>DataEndianType</name>
    <configValue>Big</configValue>
  </config>
  <config>
    <!-- algorithm's cross-granule scan requirements exclusive -->
    <!-- of any extended granule requirements -->
    <name>Crossgranule_Scans</name>
    <configValue>2</configValue>
  </config>
  <config>
    <name>Mode</name>
    <configValue>NOAUTOSSETUP</configValue>
  </config>
  <config>
    <name>Wait</name>
    <configValue>True</configValue>
  </config>
  <config>
    <name>OfficialShortName_1</name>
    <configValue>VIIRS-M2-FSDR</configValue>
  </config>
  <config>
    <name>OfficialShortNameConversionProduct_1</name>
    <configValue>VIIRS-M2-SDR</configValue>
  </config>
</group>
```

HARDCOPY UNCONTROLLED

Appendix H – Example Using NOAUTOSSETUP Mode for Outputs (from VIIRS SDR ATMSRemap Configuration Guide)

Note: only last three of five ATMSRemap outputs shown.

```
<group name="ATMS-REMAP-SDR-GEO">
  <config>
    <name>DataEndianType</name>
    <configValue>Both</configValue>
  </config>
  <config>
    <name>OfficialShortName</name>
    <configValue>ATMS-REMAP-SDR-GEO</configValue>
  </config>
</group>

<!-- Used to generate ATMS-REMAP-SDR-GEO -->
<group name="CrIS-SDR-GEO">
  <config>
    <name>DataEndianType</name>
    <configValue>Both</configValue>
  </config>
  <config>
    <name>Mode</name>
    <configValue>NOAUTOSSETUP</configValue>
  </config>
  <config>
    <name>OfficialShortName</name>
    <configValue>CrIS-SDR-RGEO</configValue>
  </config>
  <!-- Possible Output Types are DMS or HEAP -->
  <config>
    <name>OutputType</name>
    <configValue>HEAP</configValue>
  </config>
</group>

<!-- Used in the CrIMSS EDR, CrIS GIP, and CrIS ANC algorithms
      downstream -->
<group name="CrIS-SDR-GEO-TC">
  <config>
    <name>DataEndianType</name>
    <configValue>Both</configValue>
  </config>
  <config>
    <name>Mode</name>
    <configValue>NOAUTOSSETUP</configValue>
  </config>
  <config>
    <name>OfficialShortName</name>
    <configValue>CrIS-SDR-RGEO-TC</configValue>
  </config>
  <!-- Possible Output Types are DMS or HEAP -->
  <config>
    <name>OutputType</name>
    <configValue>DMS</configValue>
  </config>
</group>
```

HARDCOPY UNCONTROLLED